

I CLAIM:

1. A computer-implemented method for user interface automation synchronization, comprising:

5 sub-classing selected windows displayed on the user interface, wherein each of the selected windows corresponds to a message-queue of a thread;

setting timers corresponding to each of the selected windows;

determining whether a particular timer associate with one of the selected windows has fired; and

10 determining whether the message-queues of the threads are empty, such that when the message-queues are empty, the user interface automation proceeds to a next action.

2. The computer-implemented method of claim 1, further comprising calling a synchronization application program interface to initiate sub-classing of the selected
15 windows.

3. The computer-implemented method of claim 1, further comprising calling a wait application program interface to determine whether the message-queues of the threads are empty.

4. The computer-implemented method of claim 3, wherein message-queues
20 of the threads are indicated as empty when the user interface automation is waiting on the call to the wait application program interface.

5. The computer-implemented method of claim 1, wherein sub-classing a selected window corresponds to replacing window procedure code corresponding to the selected window with user interface automation code.

25 6. The computer-implemented method of claim 1, wherein the timers corresponding to each of the selected windows are set to zero.

7. The computer-implemented method of claim 1, wherein a determination that a particular timer associated with one of the selected windows has fired indicates that a message-queue corresponding to the one of the selected windows is empty.

8. The computer-implemented method of claim 1, further comprising setting
5 timers corresponding to system activities.

9. The computer-implemented method of Claim 8, further comprising additionally determining whether the message-queues of threads corresponding to the system activities are empty before the user interface automation proceeds to a next action.

10. A computer-readable medium that includes computer-executable
10 instructions for providing user interface automation synchronization, comprising:
hooking selected windows displayed on the user interface to correspond to user interface automation code, wherein each of the selected windows corresponds to a message-queue of a thread;
setting timers corresponding to each of the selected windows;
15 determining whether a particular timer associate with one of the selected windows has fired; and
determining whether the message-queues of the threads are empty, such that when the message-queues are empty, the user interface automation proceeds to a next action.

20 11. The computer-readable medium of claim 10, further comprising calling a synchronization application program interface to initiate sub-classing of the selected windows.

12. The computer-readable medium of claim 10, wherein hooking selected windows further comprises sub-classing selected windows such that window procedure
25 code corresponding to the selected window is replaced with user interface automation code.

13. The computer-readable medium of claim 10, wherein hooking selected windows further comprises attaching a dynamic link library to window procedure code corresponding to the selected window.

14. The computer-readable medium of claim 10, further comprising calling a
5 wait application program interface to determine whether the message-queues of the threads are empty.

15. The computer-implemented method of claim 14, wherein message-queues of the threads are indicated as empty when the user interface automation is waiting on the call to the wait application program interface.

10 16. The computer-implemented method of claim 10, wherein a determination that a particular timer associate with one of the selected windows has fired indicates that a message-queue corresponding to the one of the selected windows is empty.

17. A system for synchronizing user interface automation, comprising:
an application that includes an application program interface that is
15 configured to:

hook selected windows displayed on the user interface to correspond to user interface automation code, wherein each of the selected windows corresponds to a message-queue of a thread;

set timers corresponding to each of the selected windows;

20 determine whether a particular timer associate with one of the selected windows has fired; and

determine whether the message-queues of the threads are empty, such that when the message-queues are empty, the user interface automation proceeds to a next action.

18. The system of claim 17, wherein the application program interface is further configured to call a wait application program interface to determine whether the message-queues of the threads are empty.

5 19. The system of claim 17, wherein message-queues of the threads are indicated as empty when the user interface automation is waiting on the call to the wait application program interface.

20. The system of claim 17, wherein the application program interface is further configured to set the timers corresponding to each of the selected windows to zero.

10 21. The system of claim 17, wherein the application program interface is further configured to set timers corresponding to system activities.

15 22. The system of claim 21, wherein the application program interface is further configured to additionally determine whether the message-queues of threads corresponding to the system activities are empty before the user interface automation proceeds to a next action.